

# Easter-Eggs bei Disney und Pixar

*Tino Marquez*

Disney und Pixar sind mitunter die größten Filmstudios für Zeichentrick- und Animationsfilme. Viele Filmfiguren werden wiederverwertet usw. Daher galt unser Forschungsinteresse der Frage, in welchen Filmen treten gehäuft Cameos auf, welche Figuren kommen immer wieder vor und lässt sich eine Korrelation nach Produktionsbudgets ableiten?

Was ist ein *Cameo-Auftritt*?

NEtzwerkmaße die für uns eine Rolle Spielen etc

## Die Entwicklung des Netzwerks

Im ersten Schritt soll gezeigt werden, wie die Daten zu den wichtigsten Filmfiguren und Mitarbeitern erhoben und für die Verwendung in RStudio aufbereitet wurden.

Bei diesem Projekt entschieden wir uns zu jedem Film die wichtigsten Figuren mit zu erfassen, selbst wenn diese keine Cameo-Auftritte in anderen Filmen haben. Das hatte den Vorteil, dass eine Clusterung der Figuren nach Filmen möglich ist und sich die einzelnen Filme enger zusammenrücken. Im Gegenzug bedeutete das aber auch einen erheblichen Mehraufwand bei der Erfassung der Daten. Schlussendlich wuchs die Edgelist auf über 3000 Verbindungen.

### Quellen für das Projekt:

[https://pixar.fandom.com/wiki/Pixar\\_Wiki](https://pixar.fandom.com/wiki/Pixar_Wiki)  
[https://pixar.fandom.com/wiki/References\\_to\\_Upcoming\\_Films](https://pixar.fandom.com/wiki/References_to_Upcoming_Films)  
[https://disney.fandom.com/wiki/The\\_Disney\\_Wiki](https://disney.fandom.com/wiki/The_Disney_Wiki)  
<https://www.imdb.com/>  
<https://en.wikipedia.com>  
<https://www.gamesradar.com/50-best-pixar-easter-eggs/>  
<https://www.insider.com/pixar-movie-easter-eggs-2017-1>  
[https://en.wikipedia.org/wiki/List\\_of\\_Pixar\\_film\\_references](https://en.wikipedia.org/wiki/List_of_Pixar_film_references)

```
library(igraph) #Zur Darstellung der Netzwerke
library(igraphdata) #Um Daten auslesen zu können
library(visNetwork) #eine Alternative zu igraph, die wir nicht wirklich nutzen
library(tidygraph) #eine Weitere Alternative zu igraph,
#die großes Potential hat, aber nur schlecht dokumentiert ist.
library(ggplot2) #notwendig um mit tidygraph zu arbeiten
library(ggraph) #notwendig um mit tidygraph zu arbeiten
library(stringr) #notwendig um mit tidygraph zu arbeiten
library(rgl)
library(knitr)
library(tinytex)
library(rmarkdown)
```

## Optionen für die Arbeitsumgebung

Leider bringt ein großes Netzwerk die R-Studio Umgebung sehr schnell an Grenzen. Standardmäßig limitiert R-Studio die Ausgabe von Variablen auf etwa 400. Durch die enorme Größe unseres Netzwerks können bei den Default-Einstellungen nicht alle Verbindungen dargestellt werden. Dafür muss mit dem folgenden Befehl die maximale Anzahl an Darstellungen erhöht werden.

```
options(max.print=999999)
```

## Einlesen und Überprüfen der Daten

Da es bei unserem Projekt insgesamt über 3000 Datenpunkte gibt, haben wir uns entschieden, die Edgelisten der Figuren von den Edgelisten der Mitarbeiter (Directors, Producers und Writers), zu trennen. Rein technisch war es problemlos möglich die Netzwerke mit sämtlichen Akteuren darzustellen, allerdings ließ sich die Datenlast deutlich reduzieren. Gleichzeitig konnte auch eine, durch die größere Nodelist entstandene Verzerrung der Darstellung, durch das entfernen überflüssiger Akteure im Netzwerk vermieden werden.

Durch das Auftrennen der beiden Netzwerke ist es allerdings nicht mehr möglich zu sehen, welche Mitarbeiter der verschiedenen Studios die meisten Cameos einsetzen.

macht es Sinn hier nochmal eine gesamte Liste zu machen?

## Einlesen der Filme und Figuren

Die Edge und Nodelist der Figuren und Filme wird über read.csv von Github geladen

```
edges <- read.csv("https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/el_fig.csv", h
nodes <- read.csv("https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/nl_fig.csv", h
matrix <- as.matrix(edges) #Matrix erstellen
disnet <- graph_from_data_frame(d=matrix, vertices=nodes, directed=FALSE) #Gesamtnetzwerk wird im Dataf
disraw <- graph_from_data_frame(d=matrix, vertices=nodes, directed=FALSE) #Gesamtnetzwerk als Rohfassung
```

## Einlesen der Mitarbeiter und Filme

Als zweite Datenquelle wird auch die Edge und Nodelist der Mitarbeiter und Filme mit dem Befehl read.csv von Github geladen.

```
edgesma <- read.csv("https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/el_di.csv", l
nodesma <- read.csv("https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/nl_di.csv", l
matrixma <- as.matrix(edgesma) #Matrix erstellen
disnetma <- graph_from_data_frame(d=matrixma, vertices=nodesma, directed=FALSE) #Gesamtnetzwerk wird im
```

## Werte Überprüfen und Netzwerk validieren

Da es zu Beginn unserer Arbeiten Unstimmigkeiten zwischen der Darstellung des Netzwerks und den Daten in der Edge- und Nodelist gab, war der erste Schritt eine Überprüfung der Daten, die R-Studio in der Matrix speicherte. Für einen reinen plot der Netzwerke kann dieser Teil auch auskommentiert werden.

## Edge- und Vertex-Attribute der Figuren anzeigen

```
list.vertex.attributes(disnet)
list.edge.attributes(disnet)
```

## Edge- und Vertex-Attribute der Mitarbeiter anzeigen

```
list.vertex.attributes(disnetma)
list.edge.attributes(disnetma)
```

## Überprüfung der Node-Daten der Figuren

```
V(disnet)$name #Name des jeweiligen Knotens aus der Figuren-Modelist
V(disnet)$art # Art des jeweiligen Knotens aus der Figuren-Modelist
V(disnet)$prod # Art des jeweiligen Knotens aus der Figuren-Modelist
V(disnet)$year # Art des jeweiligen Knotens aus der Figuren-Modelist
V(disnet)$income # Art des jeweiligen Knotens aus der Figuren-Modelist
```

## Überprüfung der Node-Daten der Mitarbeiter

```
V(disnetma)$name #Name des jeweiligen Knotens aus der Figuren-NodeList
V(disnetma)$art # Art des jeweiligen Knotens aus der Figuren-NodeList
V(disnetma)$prod # Art des jeweiligen Knotens aus der Figuren-NodeList
V(disnetma)$year # Art des jeweiligen Knotens aus der Figuren-NodeList
V(disnetma)$income # Art des jeweiligen Knotens aus der Figuren-NodeList
```

Durch das händische Vergleichen zeigte sich bald, dass große Teile der Daten fehlten. Durch einige Anpassungen der Edge- und Nodelist sowie die Erweiterung des Darstellungsspeichers von R-Studio konnten nun sämtliche Daten fehlerfrei in eine Matrix überführt werden.

## Plot des Rohnetzwerks

Ausgabe des Gesamtnetzwerks der Figuren ohne grafische Anpassung

```
plot (disraw, main="Gesamtnetzwerk der Figuren")
```



Abb. 1 - Rohdaten-Gesamtnetzwerk der Figuren ohne grafische Ausgestaltung

## Statistiken zum Netzwerk

Da ein erster Plot des Gesamtnetzwerks nur einen gigantischen Hairball, also ein unübersichtliches Netzwerk aus viel zu vielen Knoten produziert (der Plot selbst kommt weiter unten), konzentrieren wir uns zunächst auf die reinen statistischen Daten. Auf diese Weise konnten wir bequem die vielversprechendsten Charaktere und Mitarbeiter selektieren, die wir in den Ego-Netzwerken untersuchen könnten.

### Figuren

#### Ausgabe der Statistiken zu den Figuren

```
disdeg <- degree(disnet, mode="IN") #Hier lässt sich der Knoten mit den meisten Verbindungen finden
View(disdeg) #Da die Console die Ausgabe auf eine gewisse Anzahl Ansgaben begrenzt, muss die Tabelle mit
components(disnet) #Components zeigt die Anzahl der Teilnetzwerke und deren Größe
mean_distance(disnet) #Gibt die längste Verbindung zwischen zwei Knoten aus
edge_density(disnet) #Gibt die Kantendichte des Netzwerks aus
diameter(disnet)
farthest.nodes (disnet)
disbetween <- betweenness(disnet)
View(disbetween)
```

Folgende Figuren wollten wir in Ego und Teilnetzwerken weiter untersuchen: A113, Buzz Lightyear, Donald Duck (15), Mickey Mouse (24), Pinocchio (9), Ron Clements, Ball from Luxo (27), Princess Jasmine (11), Tinker Bell (7)

Bei den Betweenness Werten stellte sich vor allem der Film “Ralph Breaks The Internet” als besonders interessant heraus.

### Mitarbeiter

#### Ausgabe der Statistiken zu den Mitarbeitern

```
madeg <- degree(disnetma, mode="IN") #Hier lässt sich der Knoten mit den meisten Verbindungen finden
View(madeg) #Da die Console die Ausgabe auf eine gewisse Anzahl Ansgaben begrenzt, muss die Tabelle mit
components(disnetma) #Components zeigt die Anzahl der Teilnetzwerke und deren Größe
mean_distance(disnetma) #Gibt die längste Verbindung zwischen zwei Knoten aus
edge_density(disnetma) #Gibt die Kantendichte des Netzwerks aus
```

Folgende Mitarbeiter wollten wir in Ego und Teilnetzwerken weiter untersuchen: Clyde Geronimi, Hamilton Luske, Wilfred Jackson, Walt Disney, Wolfgang Reitherman, Bill Peet, Erdman Penner, John Lasseter, Ralph Wright, Ted Sears, Vance Gerry

## Einstellungen zur grafischen Darstellung des Netzwerks

In der folgenden Darstellung zeigt sich, dass unser Gesamtnetzwerk nicht besonders übersichtlich ist.

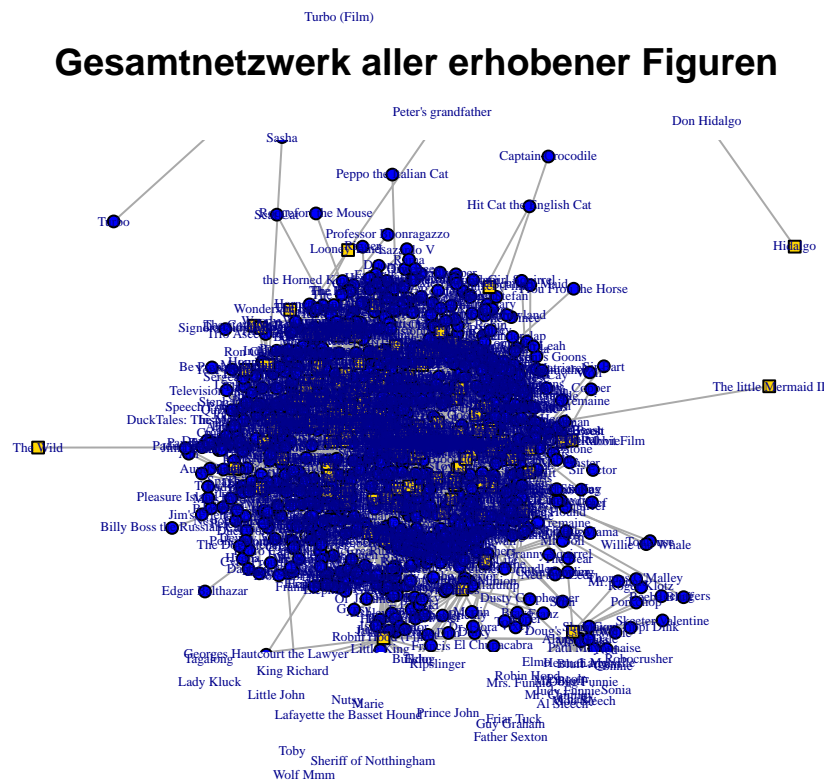
Da wir mit einem solchen Netzwerk nicht arbeiten können, ist eine Anpassung der Darstellung notwendig, mit der die einzelnen Knoten unterschieden werden können. Aufgrund der getrennten Netzwerke müssen die Einstellungen für die Farben oder Gewichtungen der Kanten jeweils doppelt vorgenommen werden. Im gleichen Schritt wird eine Entzerrung des Netzwerks versucht.

### Grafische Anpassung der Knoten und Kanten des Gesamtnetzwerks

## Disney-Pixar Gesamtnetzwerk der Figuren

```
#zuweisen des Layouts für einfache Skalierbarkeit
durchmesser <- layout.kamada.kawai(disnet)
#zuweisen der Grundkoordinaten
durchmesser <- norm_coords(durchmesser, ymin=-1, ymax=1, xmin=-1, xmax=1)

plot(disnet,
     vertex.size=5,
     vertex.label.cex=0.4,
     rescale=F,
     layout=durchmesser*1.6,
     edge.width=1,
     main="Gesamtnetzwerk aller erhobener Figuren") # erste Darstellung des Gesamtnetzwerks
```



```
#data <- toVisNetworkData(disnet)
#visNetwork(nodes = data$nodes, edges = data$edges, height = "500px")
```

Da sich in dem Gesamtnetzwerk keine Beziehungen erkennen lassen, muss das Netzwerk zunächst etwas eingedampft werden. Dazu entfernen wir sämtliche Knoten, die nur eine Verbindung aufweisen. Also Figuren, die in nur einem Film vorkommen oder Isolates sind.

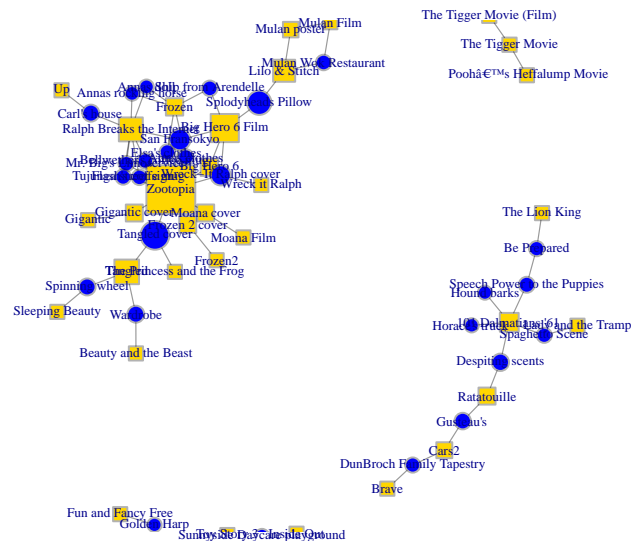


## Teilnetzwerke

### Erstes Teilnetzwerk

Hier versuchen wir ein erstes Teilnetzwerk darzustellen

```
a1 <- delete.vertices(disnet, V(disnet)[art == 2])
a2 <- delete.vertices(a1, V(a1)[art == 3])
a3 <- delete.vertices(simplify(a2), degree(a2) <= 0 )
par(mfrow=c(1,1))
durchmesser3 <- layout.kamada.kawai(a3)
durchmesser3 <- norm_coords(durchmesser3, ymin=-1, ymax=1, xmin=-1, xmax=1)
plot(a3, vertex.label.cex=0.4, layout=durchmesser3*1.1, rescale=FALSE, vertex.frame.color=rgb(0,0,0,al
      vertex.frame.width=0.8,
      edge.width=0.6,
      label.dist=1,
      edge.color=rgb(0,0,0,0.4), vertex.size=20*(betweenness(a3) / 600 + 0.3))
```



### Ego-Netzwerke spannender Akteure oder Filme

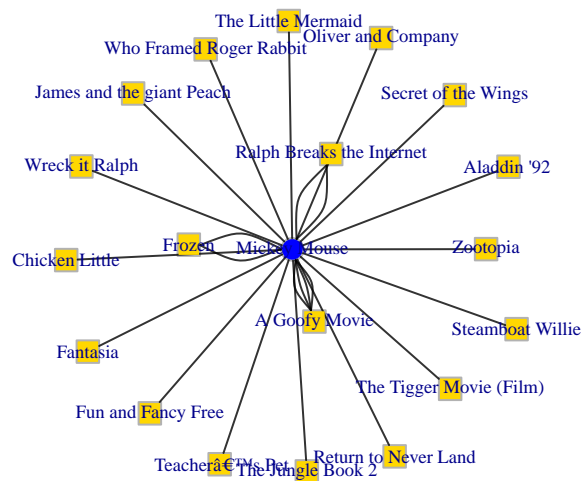
Ego Netzwerke können vielleicht auch etwas aussagen

### Mickey Mouse

spannend?



```
# Mickey Mouse
mickey_ego <- make_ego_graph(disnet, order=1, c("Mickey Mouse"))
mmego <- mickey_ego[[1]]
plot(mmego, vertex.size=10, edge.width=1, vertex.label.cex=0.6, vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.width=0.8,
     edge.width=0.6,
     label.dist=1,
     edge.color=rgb(0,0,0,0.8),)
```



## Ralph breaks the Internet

In der Voruntersuchung der Daten hat sich rbti schon als spannend herausgestellt

```
rbrks_ego <- make_ego_graph(disnet, order=2, c("Ralph Breaks the Internet"))
rbrksi <- rbrks_ego[[1]]
rbrks_egosm <- make_ego_graph(disnet, order=1, c("Ralph Breaks the Internet"))
rbrksie <- rbrks_egosm[[1]]
rbrkssm <- delete.vertices(rbrks_ego[[1]],which(degree(rbrks_ego[[1]])<2)) #Hier werden alle 1-fachen u
```

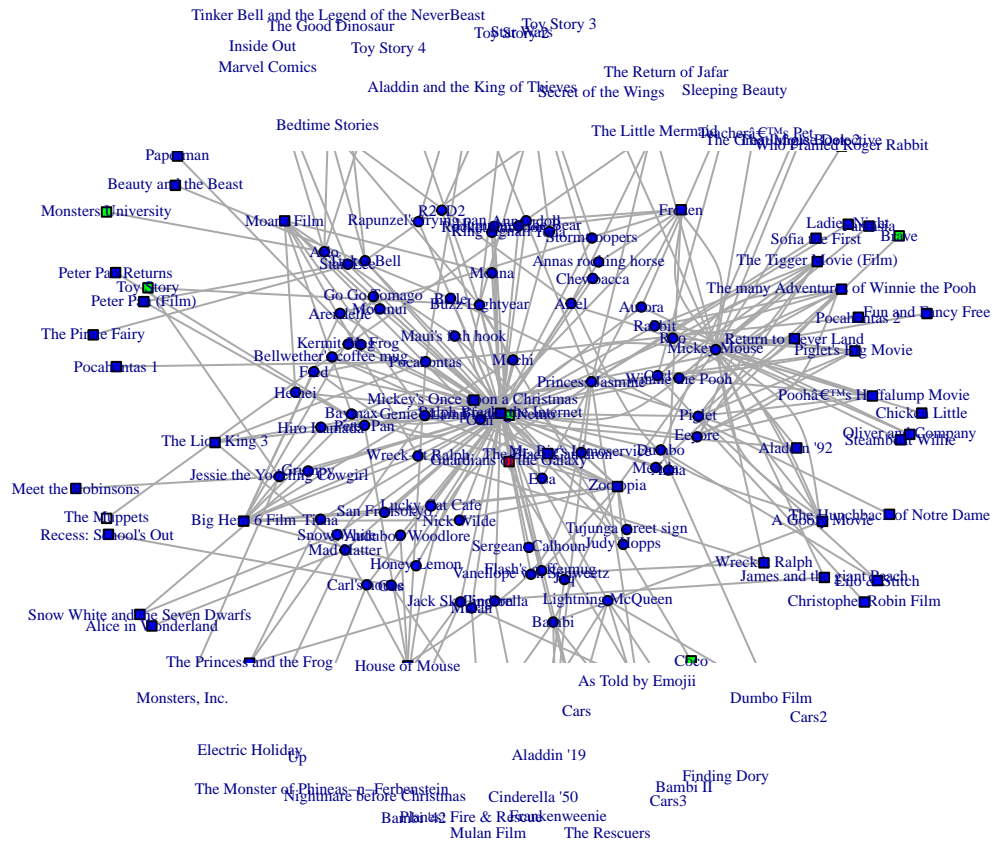
```
dmrego <- layout.kamada.kawai(rbrksi)
dmrego <- norm_coords(dmrego, ymin=-1, ymax=1, xmin=-1, xmax=1)
```

```
dmrsm <- layout.kamada.kawai(rbrkssm)
dmrsm <- norm_coords(dmrsm, ymin=-1, ymax=1, xmin=-1, xmax=1)
```



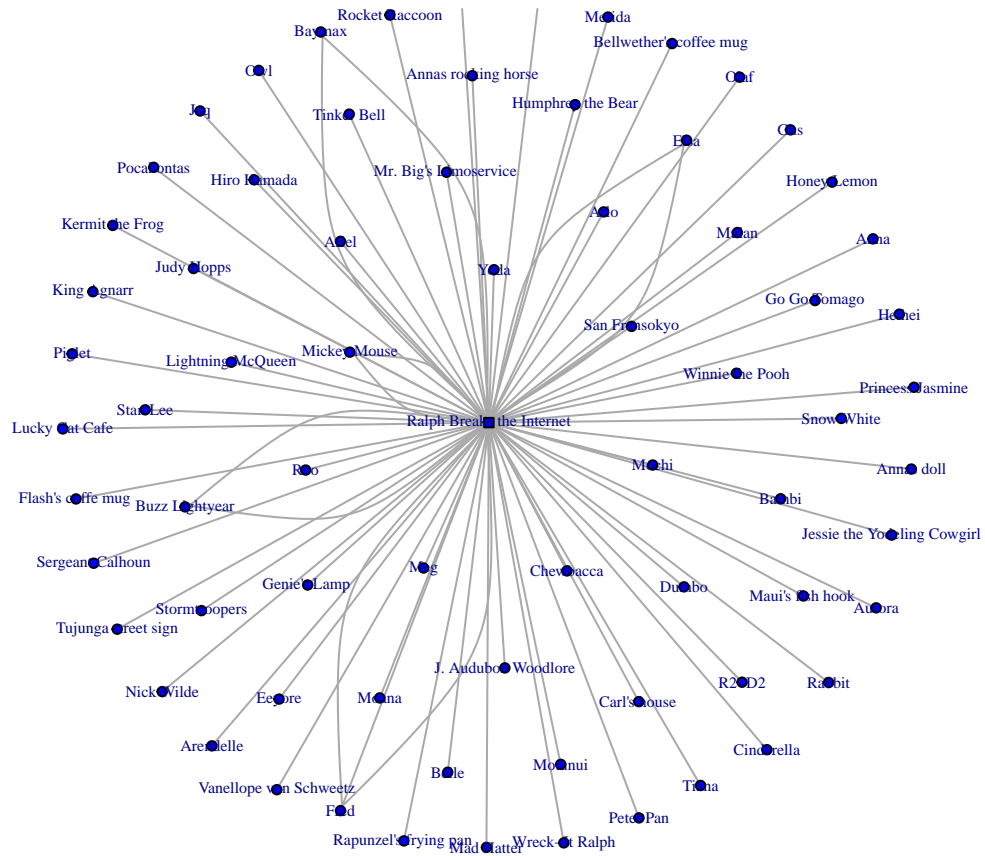
```
dmresm <- layout.kamada.kawai(rbrksie)
dmresm <- norm_coords(dmresm, ymin=-1, ymax=1, xmin=-1, xmax=1)
```

```
par(mfrow=c(1,1)) #Anzeige für zwei Netzwerke partitionieren
plot(rbrks_ego[[1]], rescale=FALSE, vertex.label.cex=0.5, edge.width=1, layout=dmrego*1.8) #Plot Ralph
```

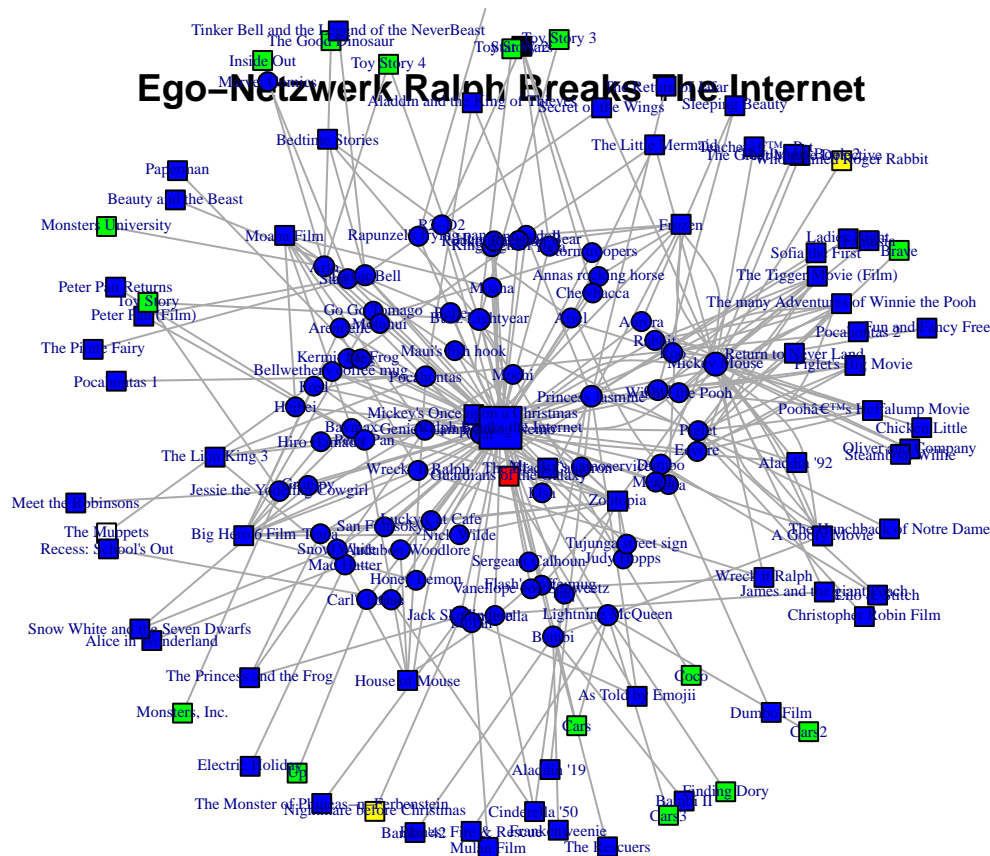


```
plot(rbrkssm, rescale=FALSE, vertex.label.cex=0.5, edge.width=1, layout=dmrsm*1.8) #Plot Ralph Ego mit
```





```
plot(rbrksi, rescale=FALSE, vertex.label.cex=0.5, edge.width=1, layout=dmrego*1.8, main="Ego-Netzwerk R
```



## Ego Netzwerke der Mitarbeiter

Clyde Geronimi, Hamilton Luske, Wilfred Jackson, Walt Disney, Wolfgang Reitherman, Bill Peet, Erdman Penner, John Lasseter, Ralph Wright, Ted Sears, Vance Gerry

```
par(mfrow=c(1,1))

#Clyde Geronimi
clyde_ego <- make_ego_graph(disnetma, order=2, c("Clyde Geronimi"))
plot(clyde_ego[[1]])

#Hamilton Luske
hamilton_ego <- make_ego_graph(disnetma, order=1, c("Hamilton Luske"))
plot(hamilton_ego[[1]])

#Walt Disney
walt_ego <- make_ego_graph(disnetma, order=2, c("Walt Disney"))
plot(walt_ego[[1]])

#John Lasseter
lasseter_ego <- make_ego_graph(disnetma, order=2, c("John Lasseter"))
plot(lasseter_ego[[1]])
```

## Beschränkung der Figuren nach den Produktionsstudios

### Cameos nach Studios - wer produziert die meisten Easter Eggs

```
pix1 <- induced.subgraph(disnet, V(disnet)[ prod %in% c("3", "") ]) #
pix15 <- delete.vertices(simplify(pix1), degree(pix1)<=2)
pix2 <- delete.vertices(simplify(pix15), degree(pix15)<=1)

pixdeg <- degree(pix2, mode="IN") #Hier können wir sehen, wie viele Cameos in welchem Film vorkamen
#View(pixdeg) #Ausgabe der Daten in einer separaten Tabelle
edge_density #Hier erfassen wir wie oft Figuren wieder verwendet werden um es mit Pixar zu vergleichen

## function (graph, loops = FALSE)
## {
##   if (!is_igraph(graph)) {
##     stop("Not a graph object")
##   }
##   on.exit(.Call(C_R_igraph_finalizer))
##   .Call(C_R_igraph_density, graph, as.logical(loops))
## }
## <bytecode: 0x000000000c68a898>
## <environment: namespace:igraph>

V(pix2)[V(pix2)$art == 1]$color <- rgb(1,0,0,alpha=0.8)
V(pix2)[V(pix2)$art == 2]$color <- rgb(0,0,1,alpha=0.8)

pixzoom <- layout.kamada.kawai(pix2) #zuweisen des Layouts für einfache Skalierbarkeit
pixzoom <- norm_coords(pixzoom, ymin=-1, ymax=1, xmin=-1, xmax=1) #zuweisen der Grundkoordinaten

dis1 <- induced.subgraph(disnet,
  V(disnet)[ prod %in% c("2", "1", "") ]) #
dis15 <- delete.vertices(simplify(dis1), degree(dis1)<=2)
dis2 <- delete.vertices(simplify(dis15), degree(dis15)<=1)

V(dis2)[V(dis2)$art == 1]$color <- rgb(1,0,0,alpha=0.8)
V(dis2)[V(dis2)$art == 2]$color <- rgb(0,0,1,alpha=0.8)
edge_density(dis2) #Hier erfassen wir wie oft Figuren wieder verwendet werden um es mit Disney zu vergl

## [1] 0.0459699

diszoom <- layout.kamada.kawai(dis2) #zuweisen des Layouts für einfache Skalierbarkeit
diszoom <- norm_coords(diszoom, ymin=-1, ymax=1, xmin=-1, xmax=1) #zuweisen der Grundkoordinaten

dip1 <- induced.subgraph(disnet,
  V(disnet)[ prod %in% c("2", "3", "1", "") ]) #
dip15 <- delete.vertices(simplify(dip1), degree(dip1)<=2)
dip2 <- delete.vertices(simplify(dip15), degree(dip15)<=2)

pixdisdeg <- degree(dip2, mode="IN") #Hier können wir sehen, wie viele Cameos in welchem Film vorkamen
#View(pixdisdeg) #Ausgabe der Daten in einer separaten Tabelle
```

```

V(dip2)[V(dip2)$art == 1]$color <- rgb(1,0,0,alpha=0.8)
V(dip2)[V(dip2)$art == 2]$color <- rgb(0,0,1,alpha=0.8)

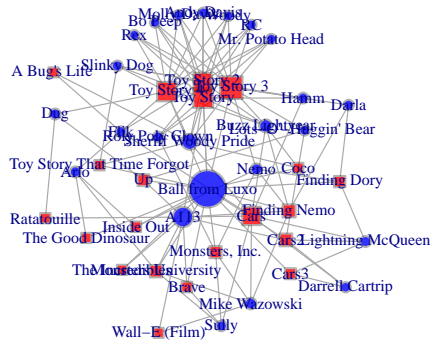
dipzoom <- layout.kamada.kawai(dip2) #zuweisen des Layouts für einfache Skalierbarkeit
dipzoom <- norm_coords(dipzoom, ymin=-1, ymax=1, xmin=-1, xmax=1) #zuweisen der Grundkoordinaten

par(mfrow=c(1,2))
plot(pix2,
     vertex.size=5000*(betweenness(pix2) / 90000 + 1/800),
     rescale=T,
     edge.width=0.6,
     vertex.label.cex=0.5,
     vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.width=0.8,
     layout=pixzoom*1.2,
     main="Cameos bei Pixar")

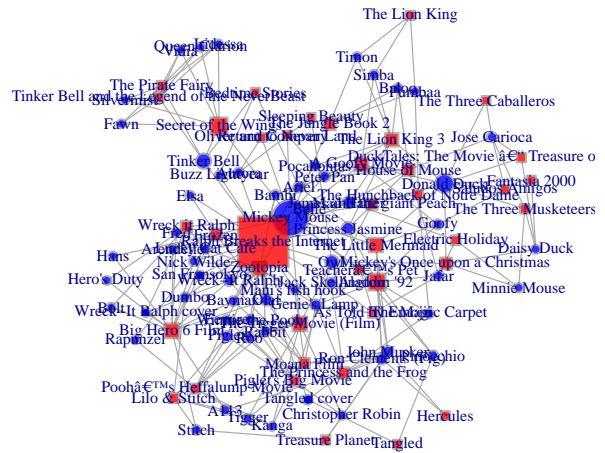
plot(dis2,
     vertex.size=1000*(betweenness(dis2) / 90000 + 1/200), #affine transformation um große Werte kleine
     rescale=F,
     vertex.label.cex=0.5,
     vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.width=0.8,
     edge.width=0.6,
     layout=diszoom*1.2,
     main="Cameos bei Disney")

```

## Cameos bei Pixar



## Cameos bei Disney

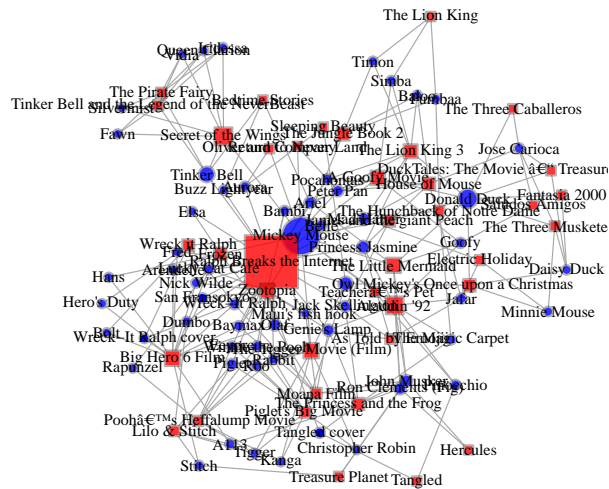


```
par(mfrow=c(1,2))
plot(dis2,
     vertex.size=1050*(betweenness(dis2) / 90000 + 1/200), #affine transformation um große Werte kleine
     rescale=F,
     vertex.label.cex=0.5,
     vertex.label.color="black",
     vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.width=0.8,
     edge.width=0.6,
     layout=diszoom*1.3,
     main="Disney Cameos Betweenness")

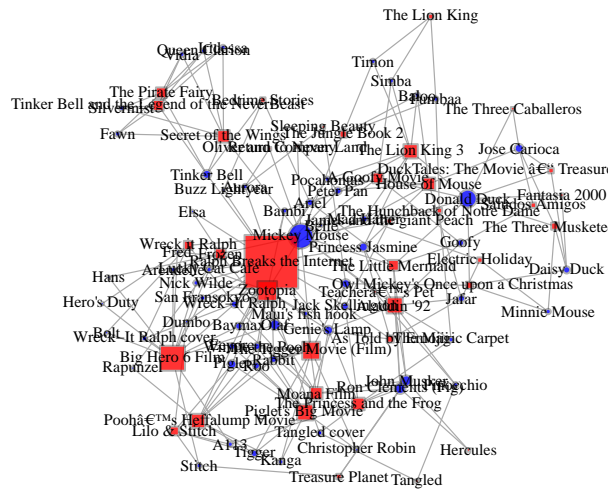
plot(dis2,
     vertex.size=degree(dis2), #affine transformation um große Werte kleiner darzustellen als sie sind
     rescale=F,
     vertex.label.cex=0.5,
     vertex.label.color="black",
     vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.width=0.8,
     edge.width=0.6,
     layout=diszoom*1.3,
     main="Disney Cameos degree")
```



## Disney Cameos Betweenness



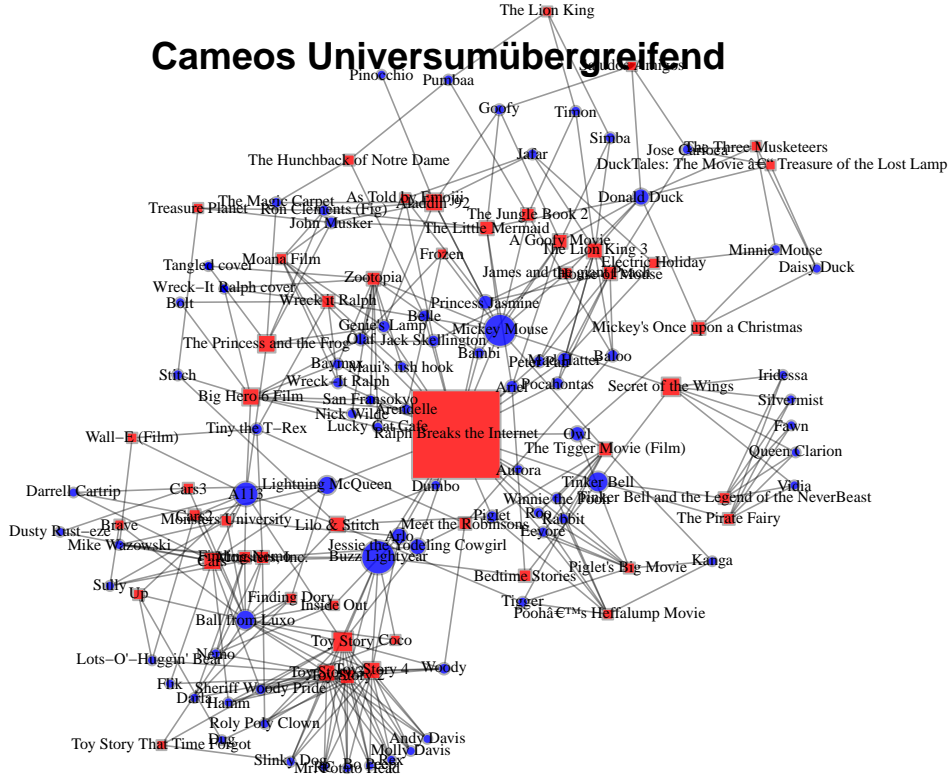
## Disney Cameos degree



```
par(mfrow=c(1,1))

plot(dip2,
     vertex.size=500*(betweenness(dip2) / 60000 + 1/120),
     rescale=F,
     edge.color=rgb(0,0,0,alpha=0.4),
     vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.width=0.8,
     edge.width=0.8,
     vertex.label.color="black",
     vertex.label.cex=0.5,
     layout=dipzoom*1.6,
     main="Cameos Universumübergreifend")
```

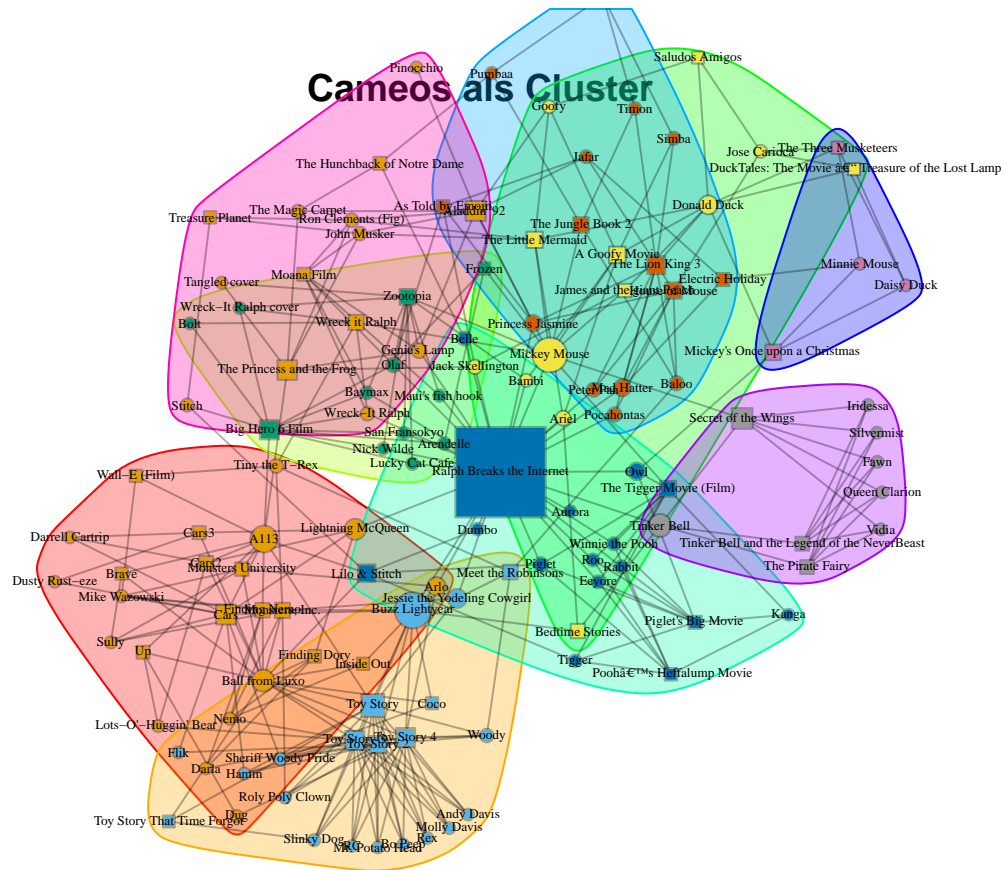
# Cameos Universumübergreifend



```
dipcluster <- cluster_label_prop(dip2) # Cluster werden gesucht
class(dipcluster)
```

```
## [1] "communities"
```

```
par(mfrow=c(1,1))
plot(dipcluster, dip2, vertex.size=1000*(betweenness(dip2) / 120000 + 1/200),
     rescale=F,
     edge.color=rgb(0,0,0,alpha=0.3),
     vertex.frame.color=rgb(0,0,0,alpha=0.3),
     vertex.label.cex=0.4,
     vertex.label.color="black",
     label.dist=1,
     vertex.label.cex=0.5,
     layout=dipzoom*1.8,
     main="Cameos als Cluster")
```



## Betweenness

Um darzustellen, welche Figuren besonders wichtig sind, greifen wir bei der Darstellung der Cameo-Netze auf den Betweenness-Wert zurück. Dieser sagt aus, welche Figuren einen besonders hohen Stellenwert im Netzwerk einnehmen, da sie verschiedene Akteure und Teilnetze überhaupt erst verbinden. Je höher dieser Wert ist, umso mehr Akteure und Teilnetze verbindet er.

Bei der Analyse unserer Teilnetzwerke (Disney und Pixar) greifen wir auf diesen Wert zurück um die Größe der Knoten in Abhängigkeit ihrer Wichtigkeit zu variieren.

## Cameos nach Studios - wer produziert die meisten Easter Eggs

### Cameos nach Mitarbeitern

### Cluster von Mitarbeitern vs. Figuren

Hier hat sich gezeigt, dass es mehr Mitarbeitercluster gibt als Figurencluster. Das bedeutet, die Figuren werden aus verschiedenen Film-Gruppen übernommen.

## Zeitliche Darstellung der Pixar Cameos

Funfact es gibt einen Film, der geteasert wurde dann aber doch nicht gedreht wurde.